

ABSTRACT OF THE DISCLOSURE

A method and apparatus for debugging of OS kernel and applications software that does not require use of a hardware probe; can debug both user-mode programs and a significant body of the OS kernel code; allows the OS to continue servicing exceptions while debugging; leverages OS built-in device drivers for communicating devices to communicate with the host debugger; and can debug a production version of the OS kernel. When debugging is required, the running OS kernel dynamically loads a software-based debug agent on demand whereby such debug agent dynamically modifies the running production OS kernel code and data to intercept debugging traps and provide run-control. To provide debugging of loadable module, the debug agent implement techniques to intercept the OS module loading system call; set breakpoints in the loaded module initialization function; calculate the start address of the debugged module in memory; and asynchronously put the system under debug. By structuring command loop to execute in non-exception mode, and devising a process to transfer execution from the debug agent exception handler to the debug agent command loop and back, the debug agent can communicate with the host debugger using interrupt-driven input/output devices as well as allowing the system to service interrupts while under debug.